Encrypt & Sync Documentation

Release 0.7.2

Ivan Konovalov

Aug 02, 2018

Contents:

CHAPTER 1

Introduction

Encrypt & Sync is a file synchronization utility with client-side encryption support.

1.1 Features

1. AES encryption support, including filename encryption
2. Synchronization between storages in any direction
3. Downloader
4. Multithreading
5. Can sync file metadata (permissions, modified date, ownership)

1.2 Supported storages

- Yandex.Disk
- Dropbox
- Local
- SFTP

Synchronization, downloading, etc. can work in any direction from any one storage to another.

This list will expand in the future.

1.3 Links

- Official website

- Documentation
- GitHub
- PyPI
- Contact

CHAPTER 2

Getting Started

2.1 Installation

2.1.1 Installing on Windows

1. Download the portable zip archive from here (64-bit) or here (32-bit)
2. Unpack it anywhere you want
3. The executables are in the `bin` directory
4. You can add the bin directory to `PATH` for convenience

2.1.2 Installing on Arch Linux

Install the package from AUR using aurman:

```
aurman -S python-eas
```

2.1.3 Installing on other systems

Installing from PyPI

Install from PyPI using pip:

```
pip install eas
```

Installing from the website

1. Download the python wheel from here

2. Install it with pip:

```
pip install eas-0.7.2-py3-none-any.whl
```

2.2 Configuration

2.2.1 Interactive configuration

To interactively edit the configuration, run

```
eas configure
```

Using the above command you can also change the master password and the encryption key.

It is recommended for you to take a look at *Concepts & Usage*, as well as `--help`:

```
eas --help
```

2.2.2 Manual configuration

The configuration file is located at `~/.eas/eas.conf`.

You can generate a sample configuration by running

```
eas make-config ~/.eas/eas.conf
```

See *Configuration*.

CHAPTER 3

Concepts & Usage

3.1 Folder

A folder refers to a directory in a certain storage. Folder path is specified like this:

```
<storage-name>:///path/to/the/folder
```

If `<storage-name>` is empty, the path will be considered local.

3.1.1 Supported storages

Storage name	Path example
Local	/local/path or local:///local/path
Yandex.Disk	disk://remote/path or yadisk://remote/path
Dropbox	dropbox:///remote/path
SFTP	sftp://user@host:22/some/path

Every folder must also have a name. Folder names must only contain the following characters:

- Letters, including unicode
- Digits
- _ (underscore), - (minus), + (plus) and . (dot).

Folders can be specified manually in the configuration file or interactively using the configure command.

3.2 Encryption

Encrypt & Sync uses AES encryption to encrypt file content and filenames. The encryption key is stored in a separate file, encrypted with the master password.

Note: If you want to change the encryption key, you'll have to re-encrypt all your existing folders that use the current key. This doesn't apply to changing the master password.

3.2.1 Filename encryption

After the filenames are encrypted they are essentially just a bunch of random bytes, so they need to be encoded.

There are several different filename encodings for that but it's very likely that you'll never need to use anything other than base32. The only possible reason for that is if you have very long filenames (over 128 characters). In that case you can take a look at all the available encodings and pick the one that suits your needs.

Note: If you have an existing folder that uses some filename encoding and you want to use a different one, you'll have to re-encrypt the entire folder.

Supported filename encodings

Filename encoding	Case-sensitive	Standard	Max. length
base64 (default)	✓	✓	160
base41			144
base32		✓	128

Note: The table assumes that the maximum unencrypted filename length is 255 bytes, which varies across different systems and cloud services.

3.3 Synchronizer

The main component that does the actual synchronization of files. It divides the work into targets, which later get divided into tasks. Sync targets require the source and the destination folders to be specified.

In order to synchronize folders, you can run:

```
eas sync <source-folder> <destination-folder>
```

See `eas sync --help` for additional information.

3.4 Scanner

Before you can sync the folders they need to be scanned first. Scanner is the component responsible for this. It's goal is to obtain the list of files a folder has. Synchronizer does this automatically, unless it's specifically told not to do that.

In order to manually scan a folder, you can run:

```
eas scan <folder1> <folder2> ...
```

See `eas scan --help` for additional information.

3.5 Downloader

In case you want to download some files (or even whole folders), there's a downloader.

In order to download something, run:

```
eas download <source-path> <destination-path>
```

See `eas download --help` for additional information.

3.6 Duplicate remover

Sometimes, if the synchronizer dies in the middle of uploading a file, it can produce file duplicates next time, thinking that the file wasn't actually uploaded. It's very rare and it only happens to encrypted folders. Duplicates are not dangerous, they just waste space.

The existence of duplicates is a consequence of using randomly-generated IVs (initialization vectors) for AES encryption of filenames. Because of this, you can have two files (or directories) with different encrypted filenames, but when you decrypt them — you get the same filename.

Fortunately, it's not hard to identify and remove them (not that you normally have to). Duplicates are identified by the scanner and removed by the duplicate remover. This is normally done automatically as a separate stage of synchronization.

To manually remove duplicates, run:

```
eas remove-duplicates <path1> <path2> ...
```

See `eas remove-duplicates --help` for additional information.

CHAPTER 4

Configuration

4.1 Syntax

The syntax is mostly similar to BASH, although, there are no variables and nested commands.

4.2 Commands

4.2.1 sync-threads

Sets the number of threads used for synchronization.

Usage:

```
sync-threads <positive-integer>
```

4.2.2 scan-threads

Sets the number of threads used for scanning.

Usage:

```
scan-threads <positive-integer>
```

4.2.3 download-threads

Sets the number of threads used for downloading.

Usage:

```
download-threads <positive-integer>
```

4.2.4 upload-limit

Sets the maximum file upload speed. `1.5m` means 1.5 MiB per second, `300k` means 300 KiB, etc.

Usage:

```
upload-limit <upload-speed>

# Examples:
upload-limit 1.3m # 1.3 MiB
upload-limit 500k # 500 KiB
upload-limit inf # no limit (infinity)
```

4.2.5 download-limit

Sets the maximum file download speed. `1.5m` means 1.5 MiB per second, `300k` means 300 KiB, etc.

Usage:

```
download-limit <download-speed>

# Examples:
download-limit 1.5m # 1.5 MiB
download-limit 500k # 500 KiB
download-limit inf # no limit (infinity)
```

4.2.6 n-retries

Sets the maximum number of retries for failing requests.

Usage:

```
n-retries <non-negative-integer>

# Examples:
n-retries 0 # disables retries
n-retries 10
n-retries 7
```

4.2.7 connect-timeout

Sets the connect timeout in seconds.

Usage:

```
connect-timeout <positive-number>

# Examples:
connect-timeout 20
connect-timeout 30
```

4.2.8 read-timeout

Sets the read timeout in seconds.

Usage:

```
read-timeout <positive-number>

# Examples:
read-timeout 15
read-timeout 25
```

4.2.9 scan-ignore-unreachable

Makes the scanner ignore unreachable files (e.g. encoding errors, denied permission, etc.). `false` is the default value.

Usage:

```
scan-ignore-unreachable [true | false]
```

4.2.10 temp-dir

Sets the temporary directory to be used instead of the default (−).

Usage:

```
temp-dir [<directory> | -]

# Examples:
temp-dir ~/my-temp-dir
temp-dir - # use the default directory
```

4.2.11 temp-encrypt-buffer-size

Sets the size for an in-memory buffer that is used for storing temporary files.

Usage:

```
temp-encrypt-buffer-size <size>

# Examples:
temp-encrypt-buffer-size 50m # 50 MiB
temp-encrypt-buffer-size 120m # 120 MiB
temp-encrypt-buffer-size 0 # disables the buffer
```

4.3 Blocks

4.3.1 exclude

This block can be used to exclude files from the synchronization. This can also speed up the scan.

Usage:

```
exclude {
    /path/to/local/dir/
    /path/to/local/file
    disk://path/to/remote/file
    dropbox://another/remote/path/
    *.py[co] # Globbing is supported too
    disk://*.py[co]
}
```

4.3.2 include

Does the opposite of the exclude block.

Usage:

```
include {
    /path/to/local/dir/
    /path/to/local/file
    disk://path/to/remote/file
    dropbox://another/remote/path/
    *.py[co] # Globbing is supported too
    disk://*.py[co]
}
```

4.3.3 targets

This block specifies the default targets to sync when the synchronizer receives `-a` (`--all`) argument.

Usage:

```
targets {
    python-local -> python-yadisk # From python-local to python-yadisk
    c++-local <- c++-yadisk # From c++-yadisk to c++-local
    folder1-local -> folder2-yadisk
    folder2-local => folder1-yadisk
    folder3-local folder3-yadisk
}
```

4.3.4 folders

This block is used to specify folders. Folder name must only contain letters, digits, _, -, + and `.`.

Usage:

```
folders {
    <folder-name> <folder-path> {
        encrypted [true | false] # Enable/disable folder encryption, (false by␣
↪default)
        avoid-rescan [true | false] # If true, makes the synchronizer avoid␣
↪rescanning the folder, unless it's empty in the database
        filename-encoding [base64 | base41 | base32] # Filename encoding to use for␣
↪encrypted filenames (base64 by default)
```

(continues on next page)

(continued from previous page)

```
        # Exclude certain files/directories, you can have multiple exclude/include␣
↪blocks
        exclude {
            # relative paths
        }

        # Overrides exclude blocks, you can have multiple exclude/include blocks
        include {
            # relative paths
        }
    }

    <folder-name> <folder-path> {}

    ...
}

# Examples:
folders {
    python-local ~/Python {
        exclude {
            ./some/relative/path/*
            /some/absolute/path/*
            unwanted-file
            unwanted-directory/
            .*.sw[a-z]
            *.py[co]
        }

        include {
            ./some/relative/path/do-not-exclude/*
            /some/absolute/path/do-not-exclude/*
        }
    }

    python-yadisk disk://Python {
        encrypted true
        avoid-rescan true
        filename-encoding base64
    }

    remote-folder disk://SomeFolder {
        encrypted true
    }

    some-other-folder dropbox:///some/other/folder {
        encrypted true
        filename-encoding base32
    }
}
```

CHAPTER 5

Indices and tables

- genindex
- modindex
- search